**What is claimed is:**

1. A method of initializing a computer system equipped with a debugging system, wherein the computer system has a CPU, a local, peripheral and expansion bus, a first and second bridge, and a ROM coupled to the expansion bus and storing a first BIOS code, and the debugging system is coupled to the peripheral bus, the method comprising the steps of:

routing data requests directed to the ROM to the local bus via the CPU;

transferring the data requests from the local bus to the peripheral bus via the first bridge;

switching the second bridge into a normal mode wherein the second bridge is enabled to respond to data requests on the peripheral bus with the first BIOS code in the ROM to be loaded in the CPU; and

switching the second bridge into a debugging mode wherein the second bridge is disabled from responding to data requests on the peripheral bus, instead, the debugging system responds to data requests with the second BIOS code to be loaded in the CPU.

2. The method as claimed in claim 1, wherein the second BIOS code is programmed by the debugging system.

3. The method as claimed in claim 1, wherein the debugging system comprises:

an interface card coupled to the peripheral bus; and

a second computer system coupled to the interface card.

4. The method as claimed in claim 1 further comprising the step of:

when the second bridge is switched to debugging mode, retrieving and displaying contents of registers in the CPU via the debugging system.

5. The method as claimed in claim 1 further comprising the step of:

when the second bridge is switched to debugging mode, reading the first BIOS code from the ROM by the debugging system through the second bridge.

6. The method as claimed in claim 1 further comprising the step of:

when the second bridge is switched to debugging mode, overwriting the first BIOS code in the ROM with the second BIOS code by the debugging system through the second bridge.

7. The method as claimed in claim 1, wherein switching between the normal and debugging mode is performed by a strapping pin of the second bridge.

8. The method as claimed in claim 1, wherein the peripheral and expansion bus are a PCI and an ISA bus, and the first and second bridge are a north and south bridge, respectively.

9. The method as claimed in claim 1, wherein the second bridge responds the first data requests by sending a

device select signal to the peripheral bus, decoding addresses carried in the first data requests and retrieving the first BIOS code in the ROM corresponding to the addresses.

10. The method as claimed in claim 1, wherein the debugging system responds to the second data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the second data requests and retrieving the second BIOS code therein corresponding to the addresses.

11. A computer system capable of being initialized by a debugging system, comprising:

a CPU;

a local, peripheral and expansion bus, wherein the CPU routes data requests to the local bus and the debugging system is coupled to the peripheral bus;

a ROM coupled to the expansion bus and storing first BIOS code, wherein the data requests are directed to the ROM;

a first bridge transferring the data requests from the local to the peripheral bus; and

a second bridge switched between a normal mode wherein the second bridge is enabled to respond to data requests on the peripheral bus with the first BIOS code in the ROM to be loaded in the CPU and a debugging mode wherein the second bridge is disabled from responding to data requests on the

peripheral bus, instead, the debugging system responds to data requests with the second BIOS code to be loaded in the CPU.

12. The computer system as claimed in claim 11, wherein the second BIOS code is programmed by the debugging system.

13. The computer system as claimed in claim 12, wherein the debugging system comprises:
an interface card coupled to the peripheral bus; and
a second computer system coupled to the interface card.

14. The computer system as claimed in claim 11, wherein the debugging system retrieves and displays contents of registers in the CPU when the second bridge is switched to debugging mode.

15. The computer system as claimed in claim 11, wherein the debugging system reads the first BIOS code from the ROM through the second bridge when the second bridge is switched to debugging mode.

16. The computer system as claimed in claim 11, wherein the debugging system overwrites the first BIOS code in the ROM with the second BIOS code through the second bridge when the second bridge is switched to debugging mode.

17. The computer system as claimed in claim 11, wherein the second bridge has a strapping pin for switching between normal and debugging mode.

18. The computer system as claimed in claim 11, wherein the peripheral and expansion bus are a PCI and ISA bus, and the first and second bridge are a north and south bridge, respectively.

19. The computer system as claimed in claim 11, wherein the second bridge responds to data requests by sending a device select signal to the peripheral bus, decoding addresses carried in the data requests and retrieving the first BIOS code in the ROM corresponding to the addresses.

20. A peripheral/expansion bus bridge in a computer system capable of being initialized by a debugging system, wherein the computer system further comprises a CPU, a local, peripheral and expansion bus, a local/peripheral bus bridge, and a ROM coupled to the expansion bus to store a first BIOS code, and the debugging system is coupled to the peripheral bus, the peripheral/expansion bus bridge comprising:

a means for switching the peripheral/expansion bus bridge between a normal and debugging mode; and

a decoder enabled to decode addresses carried by data requests of the CPU directed to the ROM, whereby the peripheral/expansion bus bridge sends a device select signal to the peripheral bus and retrieves the first BIOS code in the ROM corresponding to the addresses to be loaded in the CPU when the peripheral/expansion bus bridge is switched to normal mode, and disabled when the

peripheral/expansion bus bridge is switched to debugging mode;

wherein, the debugging system instead responds to data requests with the second BIOS code to be loaded in the CPU when the peripheral/expansion bus bridge is switched to debugging mode.